Guo, Muyang
Wang, Yibo
Yu, Su

Programming Assignment 3

CSE 6220
April 23, 2020

Introduction

In this programming assignment, Jacobi's method is acieved using parallel algorithm using 2-dimensional mesh as the communication network. The program is coded in C++ with MPI commands. This document describes the algorithem design, and reports the runtime and speedups achieved at different n and p.

Design Description

This parallel algorithm solves matrix computations using Jacobi's method using the following steps.

1. Create 2D Cartesian grid for the processors on matrix A, so that the rows and columns distributed to each processor are $\lceil \frac{n}{q} \rceil$ or $\lfloor \frac{n}{q} \rfloor$. A sample grid for an 8x8 matrix A with 3x3 processors are illustrated in Figure 1.

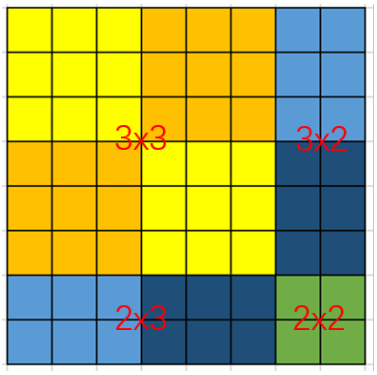

Figure 1: mesh

2. Distribute vectors b and x to the local processor as instructed by the programming assignment using Many-to-Many communicator.

3. Broadcast diagonal terms d_i from $P(i,i)$ to $P(i,j)$ using one-to-all broadcast.

4. Replace diagonal terms in $P(i,i)$ with 0 to obtain matrix R. Compute $\frac{R_{i,j}x_j}{d_j}$ locally in each processor.

5. Calculate $\sum_{i=0}^{\frac{n}{q}} \frac{R_{i,j}x_j}{d_j}$ from local results generated in previous step. This is performed for each column using parallel all-to-one reduction with summation as the operator.

6. Compute new x_j as $x_j = \frac{b_i}{d_i} - \sum_{i=0}^{\frac{n}{q}-1} \frac{R_{i,j}x_j}{d_j}$.

7. Broadcast x_j to $P(i,j)$ using one-to-all broadcast.

8. Repeat steps 4-7 until convergence.

Results and Discussion

1. Runtime at fixed n, different p

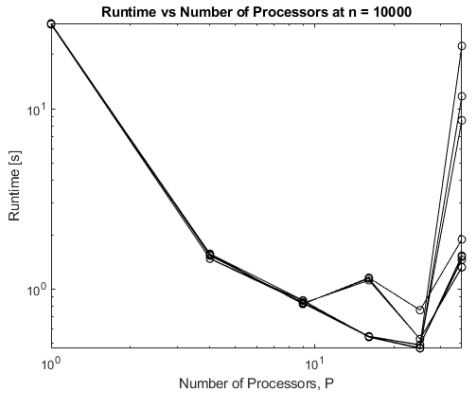

Figure 2: Runtime vs. P at $N = 10^4$

Runtime decreases with number of processors increases at first, then increases with the number ofprocessors. This is because when number of processors reaches certain amount, the communication time will dominate

2. Runtime at fixed p, different n

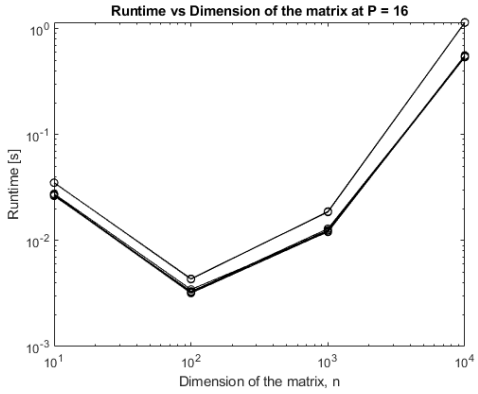

Figure 3: Runtime vs. N at $P = 16$

In general, the runtime increases with an increase in number of processors. This increase was negative at n=10, because the number of processors is greater than the number of elements in matrix, which introduces extra communication work.

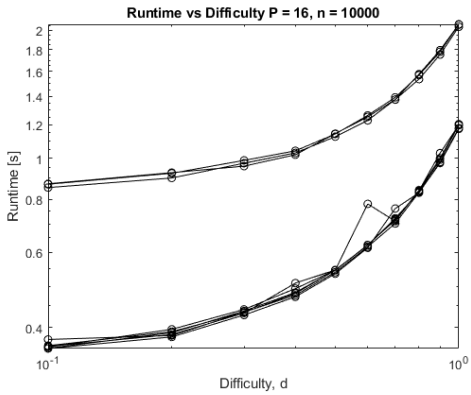

Figure 4: Runtime vs. Difficulty at $P = 16$, $N = 10^4$

In general, the runtime increases with difficulty. This increase follows a non-linear relationship. At a small difficulty, the increase was insignificant. As the difficulty level increases, the increase in runtime becomes larger and larger.

Summary

In summary, the parallel Jacobi method is studied and implemented in the project.